ICS 53, Winter 2021

Assignment 1

Problem 1

Write a program which simulates the rolling of two six-sides dice 100 times. The rolling of each die should be simulated using the rand() function which is in the stdlib.h library. Each roll should produce a number between 2 and 12. Have the program print a histogram of the number of times each value occurs.

The program should print one row for each value between 2 and 12. Each row should contain three things:

1. The rolled value (2-12)
2. The number of times, n, that the value was rolled
3. n asterisks to depict the histogram. For example, if there are 28 rolls of 7, the output line ``7: 28'' should also contain a horizontal row of 28 asterisks.

The output might look like this:

```
2: 2    **
3: 5    *****
4: 4    ****
5: 10   **********
6: 15   ***************
7: 28   ****************************
8: 12   ************
9: 9    *********
10: 7   *******
11: 5   *****
12: 3   ***
```

Name your program's source code file "p1.c".

Problem 2

Write a program which tokenizes a string entered by the user.

The program should first accept a string typed by the user. The program should then print each token on a separate line, and then print the entire string.

Remember that tokens are separated by whitespace, either blank spaces or tab characters.

For example, if the user types “Nice day, my dear.” Then your program should print the following:

```
Nice
day,
my
dear.
Nice day, my dear.
```

Name your program’s source code file “p2.c”.

Problem 3

Write a program which allows a user to maintain a calendar of events in a single day. Your program should allow the user to enter a series of commands which modify and display the calendar. Your program should print out a prompt to indicate that the user can enter a command. Your prompt should be the string ‘$ ’. At the prompt the user will type a command followed by a set of arguments for the command. When the user enters a command, your program should respond to the command appropriately and then print another prompt to indicate to the user that he/she can enter a new command. This should continue until the user enters ‘quit’ which should cause your program to end.

Your program should accept the following commands:

- **add <event_name> <start_time> <end_time>** - The add command adds a new event into the calendar. <event_name> is the name of the event. <start_time> is the start time of the event. <end_time> is the end time of the event. <event_name> may not contain blank space characters. <start_time> and <end_time> must be integers between 0 and 23 to indicate an hour on a 24 hour clock. If the time of the new event overlaps with the time of an existing event, the new event should not be added and the message “Event overlap error” should be printed.
- **delete <event_name>** - The delete command deletes an event from the calendar.
- **printcalendar** – This command prints all of the events in the calendar in order by start time.
- **quit** – This command ends your program

The following is an example of the execution of the calendar program. User inputs are highlighted in bold.

```
$ add e1 7 9
$ add e2 3 5
$ printcalendar
e2 3 5
e1 7 9
$ add e3 4 6
```

```
Event overlap error
$ delete e1
$ printcalendar
e2 3 5
$ quit
```

Name your program's source code file "p3.c".

Problem 4

Use the gcov utility to determine the line coverage produced by executing the program that you wrote for Problem 3. Execute the program and supply it with a single input, "add e1 7 9". Use gcov to compute the statement coverage and to produce the "p3.c.gcov" file. Submit the "p3.c.gcov" file.

Submission Instructions

You will upload your solutions to Gradescope. Upload 3 different files, one for each problem. Name the files p1.c (Problem 1), p2.c (Problem 2), p3.c (Problem 3), and p3.c.gcov (Problem 4). Please remember that each C program should compile and execute properly on openlab.ics.uci.edu when it is compiled using the gcc compiler version 4.8.5. The only compiler switches which will be used are -fprofile-arcs, -ftest-coverage (to allow the use of gcov), and -o (to change the name of the executable.